Давайте представим ситуацию, что у нас есть Excel файл, в котором находится обширное количество данных. И эти данные необходимо разделить по отдельным файлам, а затем отправить по электронной почте. Чтобы было более понятно, например, у меня есть файл, в котором я храню данные о клиентах. И по каждому клиенту мне нужно создать собственный файл, а затем отправить ему это по электронной почте. Пример файла изображен внизу:

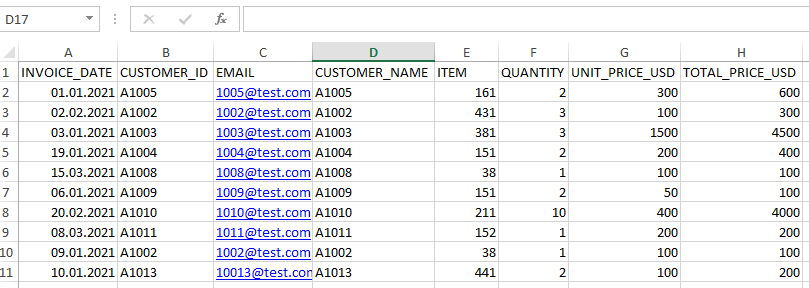


Рисунок 1 – Пример исходного документа

Если проводить эту операцию вручную, то придется потратить достаточное количество своего времени, а если файл будет содержать информацию не о 10 клиентах, как в моем случае, а о 1000, то это уже будет огромной проблемой. Но ее можно будет решить с помощью Python. Первым делом, как всегда, необходимо выполнить импорт библиотек (некоторые из них необходимо будет установить через pip):

import datetime

import os

import shutil

from pathlib import Path

import pandas as pd

import win32com.client as win32

Далее, в папке с проектом мы создаем несколько папок, а именно: archive и data, в которую кладем файл с исходными данными, в моем случае – customers.xlsx и в ней же создаем папку attachments, в которую у нас будут сохраняться наши новые файлы. После этого пишем следующий код:

## Установить формат даты

today\_string = datetime.datetime.today().strftime('%m%d%Y\_%I%p')

today\_string2 = datetime.datetime.today().strftime('%b %d, %Y')

## Установка имен папок для вложений и архивирования

attachment\_path = Path.cwd() / 'data' / 'attachments'

archive\_dir = Path.cwd() / 'archive'

src\_file = Path.cwd() / 'data' / 'customers.xlsx'

Давайте посмотрим на основной файл данных, который нужно обработать и который является источников множества файлов для клиентов:

df = pd.read\_excel(src\_file)

df.head()

Следующий шаг — сгруппировать все транзакции, относящиеся к клиентам по их CUSTOMER\_ID, вместе. Начнем с создания группы по CUSTOMER\_ID:

customer\_group = df.groupby('CUSTOMER\_ID')

for ID, group\_df in customer\_group:

print(ID)

Результат:

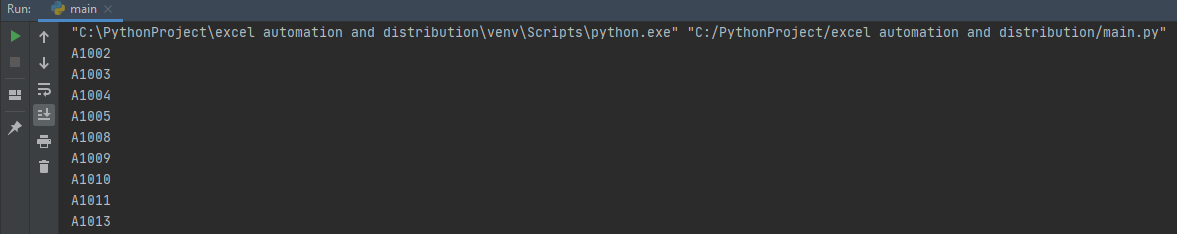


Рисунок 2 – группировка по CUSTOMER\_ID

После этого, у нас есть все необходимое, чтобы создать файл Excel для каждого клиента и сохранить его в каталоге attachments для дальнейшего использования:

## Запишите каждый идентификатор, группу в отдельные файлы Excel и используйте идентификатор,

## чтобы назвать каждый файл с сегодняшней датой

attachments = []

for ID, group\_df in customer\_group:

attachment = attachment\_path / f'{ID}\_{today\_string}.xlsx'

group\_df.to\_excel(attachment, index=False)

attachments.append((ID, str(attachment)))

df2 = pd.DataFrame(attachments, columns=['CUSTOMER\_ID', 'FILE'])

Результат:

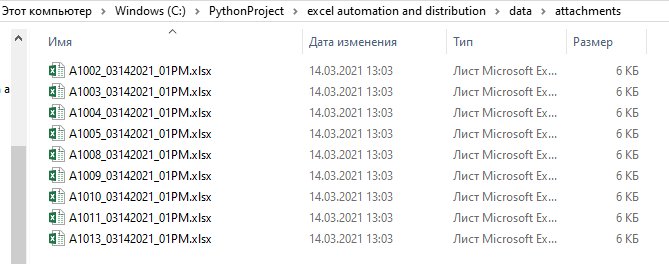


Рисунок 3 – Создание отдельных файлов по каждому клиенту

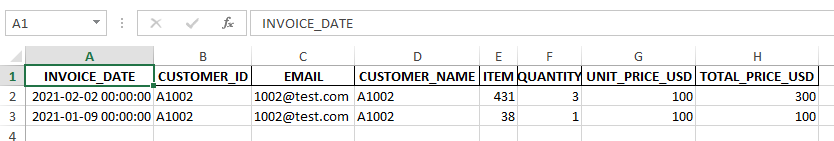


Рисунок 4 – Пример файла по одному из клиентов

В ходе всех этих проделанных действий мы собрали список клиентов с их адресами электронной почты и именами прикрепляемых файлов. Теперь нам нужно отправить электронное письмо с помощью почтового клиента. Лично я, буду использовать Outlook.

# Отправка индивидуальных отчетов по электронной почте соответствующим получателям

class EmailsSender:

def \_\_init\_\_(self):

self.outlook = win32.Dispatch('outlook.application')

def send\_email(self, to\_email\_address, attachment\_path):

mail = self.outlook.CreateItem(0)

mail.To = to\_email\_address

mail.Subject = today\_string2 + ' Report'

mail.Body = """Please find today's report attached."""

mail.Attachments.Add(Source=attachment\_path)

# Показать электронную почту

mail.Display(True)

# Отправка

mail.Send()

Также, можно использовать класс EmailsSender, который позволит создавать электронные письма и прикреплять файл Excel:

email\_sender = EmailsSender()

for index, row in combined.iterrows():

email\_sender.send\_email(row['EMAIL'], row['FILE'])

Результат:

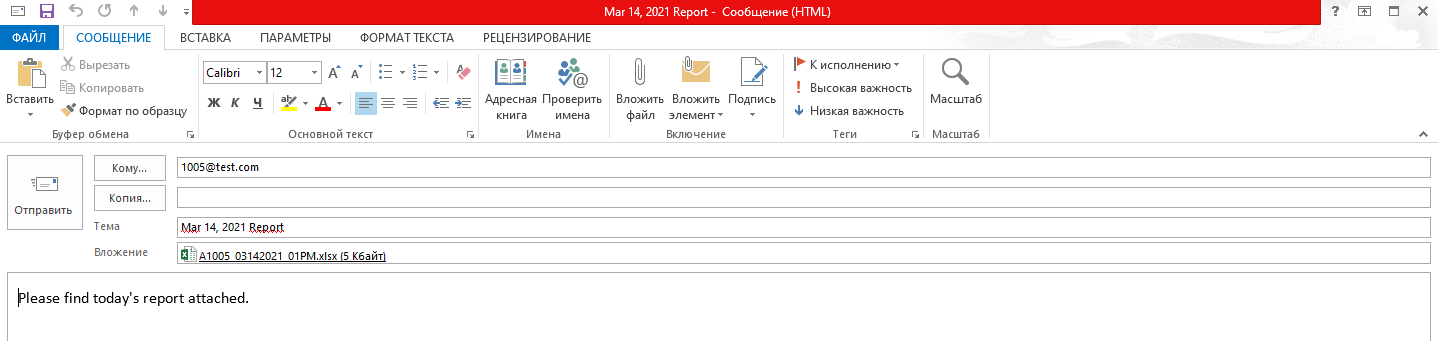


Рисунок 5 – Отправка письма

Ну и завершающим этапом будет перемещение всех этих созданных файлов в архивный каталог:

# Переместить файлы в архив

for f in attachments:

shutil.move(f[1], archive\_dir)

Благодаря данной программе, мы смогли автоматизировать достаточно болезненную ручную работу и сэкономить просто огромное количество своего времени и сил.